

REMARKS

Claims 1-6 and 17-22 are pending after this amendment.

Applicants have amended claims 1-2, 4-6, 17-18, and 20 in order to more particularly define the invention. The amendments were not necessitated by the claim rejections. Applicants make no admission as to the patentability or unpatentability of the originally filed claims.

Claims 7 to 16 were previously canceled.

Claims 21-22 have been added merely to more particularly define the invention.

The amendments and remarks presented herein are in response to the Final Office Action dated March 4, 2009.

The Examiner rejected claims 1-6 and 17-20 under 35 USC 102 as allegedly being anticipated by Bean. This rejection is respectfully traversed.

Claim 1, as amended, recites:

“A method for customizing website traffic tracking data comprising the steps of:
receiving user input specifying configuration of a web page to track at least one custom event, the user input comprising:
selection of at least one of a plurality of customizable events; and
selection of at least one custom attribute associated with at least one customizable event;
inserting embedded tracking code in the web page;
modifying the embedded tracking code to track the occurrence of the at least one custom event as specified by the user input; and

receiving user input specifying configuration of a data collection server to receive the at least one custom event, the user input specifying values for at least one event and at least one custom attribute;
responsive to the user input:
 configuring the data collection server to receive the at least one custom event;
 and
 generating a configuration string for inclusion on the web page, the configuration string comprising encoded configuration information for at least one custom attribute.”

The claim has been amended to more particularly define the invention. Specifically, the claim recites a method for customizing website traffic data. User input is received, specifying configuration of a web page to track at least one custom event. The user input comprises selection of at least one customizable event, and selection of at least one custom attribute associated with at least one customizable event. Embedded tracking code is inserted in the web page. The embedded tracking code is then modified to track the occurrence of the at least one custom event as specified by the user input. User input is received, specifying configuration of a data collection server to receive the at least one custom event; the user input specifies values for at least one custom event and at least one custom attribute. Responsive to the user input, the data collection server is configured to receive the at least one custom event, and a configuration string is generated for inclusion on the web page. The configuration string comprises encoded configuration information for at least one custom attribute.

The claimed method thus provides a novel mechanism for customizing website traffic tracking data, by providing the ability for a user to customize the event

tracking to be performed with respect to a web page. Tracking code embedded on a web page is modified to track specific custom events specified by a user; thus the invention is not limited to predefined events. The use of a configuration string generated in response to user input provides a mechanism for encoding information about one or more custom attributes; the configuration string can then be included on the web page so as to enable tracking of the specified attribute(s).

Bean fails to teach or suggest the recited limitations. Bean describes a method for tracking and reporting traffic activity on a web site, including sampling the traffic so as to reduce the resources necessary to track and report web page traffic. However, Bean merely describes tracking of predefined events, and does not provide any mechanism for inserting tracking code containing a custom event.

Nowhere in Bean is there any mention of receiving user input for selecting one of a plurality of customizable events, or for selecting custom attributes associated with such events. Furthermore, Bean fails to teach any mechanism for modifying embedded tracking code to track occurrence of a custom event. In Bean, a subscriber copies and pastes data mining code onto each web page for which monitoring is desired. (See col. 2, line 50-54). However, Bean does not teach modifying the data mining code to track the occurrence of a custom event as specified by user input, as claimed herein. In fact, Bean explicitly recites that the code passes predetermined information from the computer to a server. Col. 2, lines 54-57.

Furthermore, Bean fails to teach any technique by which a data collection server is configured, responsive to user input, to receive at least one custom event. Nor is there any teaching of a technique for generating a configuration string, responsive to user input. In the claimed method, the configuration string comprises encoded configuration information for at least one custom attribute. Nowhere in Bean is there any such teaching. Bean also fails to describe any configuration string for inclusion on a web page.

The claim, as amended, now specifically states that, once embedded tracking code has been inserted, the inserted embedded tracking code is modified to track the occurrence of the at least one custom event as specified by the user input. Nowhere in Bean is there any such teaching of the tracking of a custom event in this manner, nor of any technique of modifying tracking code to perform such tracking.

In response to Applicants' earlier argument that Bean fails to teach inserting tracking code containing a custom event, the Examiner stated that Bean discloses the process wherein data mining code is embedded within a web page script. The Examiner cited col. 3, lines 7-8 as containing this teaching. However, the cited portion of Bean merely states, "The data mining code embedded within the web page script operates to gather data about the visitor's computer." Nowhere in this statement is there any mention of a custom event, as claimed herein.

Additionally, in response to Applicants' earlier argument that Bean fails to teach modifying embedded tracking code to track occurrence of a custom event, the

Examiner stated that Bean discloses the web server changing name-value pairs or adding new pairs. The Examiner stated that Bean teaches the ability to change content/layout/color of a site and allows for customization by modification of name-value pairs. The Examiner cited col. 4, lines 22-36 and col. 5, lines 6-15 as containing this teaching.

A thorough reading of the cited portions of Bean reveals that they are completely unrelated to the claimed limitation of modifying embedded tracking code to track occurrence of a custom event. As specifically and explicitly stated in Bean, "A name-value pair is simply a named piece of data. It is not a program, and it cannot 'do' anything." Col. 4, lines 20-21. Bean also defines name-value pairs at col. 3, lines 31-25: "Cookies allow a web site to store information on a user's machine and later retrieve it. The pieces of information are stored as 'name-value pairs' comprised of, for instance, a variable name ... and a value ... associated with that variable name." The portion of Bean cited by the Examiner (col. 4, lines 22-36) simply describes the well-known process of a browser requesting a web page in response to the user typing in a URL, and the browser sending name-value pairs found in a cookie associated with the requested web page. At col. 4, lines 45-46 Bean states that the web server can change name-value pairs or add new pairs when the web page is visited; this merely sets forth the well known mechanism by which web servers can store information in cookies at a browser, including setting new name-value pairs and updating existing ones. None of these techniques have anything to do with modifying

tracking code on a web page; in fact the stored cookies (containing the name-value pairs) are completely distinct from and unrelated to any tracking code appearing on a web page. In addition, the name-value pairs discussed by Bean contain information about the user and/or other state information, as is well known in the art; thus, modifying, updating, or adding to such information is entirely different than modifying tracking code for tracking custom events.

It is important to emphasize that the claimed method explicitly recites the modification of tracking code. Tracking code is software code, equivalent to a software program, that performs a function: it tracks events. By contrast, name-value pairs contain data; as made clear by Bean, they are not programs and do not “do” anything. Modification of tracking code (or any software code, for that matter) is an entirely different operation than modifying data, and serves a completely different purpose.

Col. 5, lines 6-15 was also cited by the Examiner as containing this teaching. This portion of Bean is even more off-point. The entire passage relates solely to the ability of a website to store user preferences so as to customize the appearance for each visitor. The passage contains a description by which a ZIP code can be entered so that the user can receive customized weather information; location data is determined from the ZIP code and entered in a name-value pair.

Storage of such user information in a cookie, in name-value pair format, is well-known, ubiquitous, and quite useful. However, it is completely unrelated to the

claim language of the present application. Again, claim 1 explicitly recites modification of embedded tracking code. It further recites that such modification is done to track occurrence of a custom event specified by user input. The modification of tracking code for such a purpose is a specific operation performed by the claimed method, and is entirely distinct from Bean's conventional storage of customization or location information in a cookie, in name-value pair format. The two operations are different in nature and in purpose. It cannot be said, by any reasonable reading of Bean, that Bean's teaching in this regard anticipates the claimed limitation in any way.

In response to Applicants' earlier arguments concerning Bean, the Examiner further stated that "Bean discloses wherein the web pages having embedded code, the code passes predetermined information from computer to server and where this information includes e.g. the page viewed, the time of the view the length of the stay on the page, the visitor's identification, etc. (col. 2, lines 50-65). The Examiner considers the different types of embedded information to represent unique or custom information for each particular user or visitor of the web page." Final Office Action, p. 11.

Applicants respectfully point out that Bean explicitly states that the information passed by its code is "predetermined" information. By contrast, claim 1 explicitly recites steps for tracking occurrence of a custom event. The specific steps include selection of one (or more) customizable events from a plurality, selection of one (or

more) custom attributes, insertion of tracking code, and modification of the embedded code to track the custom event. Also recited are specific steps for receiving user input for configuring a data collection server, and generation of a configuration string. All of these steps set forth a defined and specific methodology for tracking custom events, none of which appears in Bean. The cited portion of Bean (col. 2, lines 50-65) merely sets forth well known and conventional mechanisms by which tracking code is copied and pasted onto web pages, and wherein the tracking code passes predetermined information when a web page is loaded. It cannot be reasonably said that the mere fact that different types of information can be collected via such a scheme makes the scheme equivalent to the specific mechanisms recited herein for tracking custom events. In fact, the techniques described by Bean are substantially equivalent to those discussed as background material in the specification of the present application, and thereby suffer the same deficiencies as described in the background section of the present application.

The Examiner further stated that “Bean discloses wherein cookies store unique ID, user preferences, and tracks *[sic]* user selections (col. 4, line 45 – col. 5, line 31).” Final Office Action, p. 11. Again, and as stated above, such conventional use of cookies to store such information is entirely unrelated to the specific steps for tracking custom events recited herein.

The Examiner further stated that “Bean discloses the process of offering the visitor the ability to change content, layout or color of a site and allows for customi-

zation in which will [sic] modify a name-value pair (col. 5, line 6-15).” Final Office Action, p. 11. As discussed above, this customization capability is completely unrelated to the tracking system described herein.

The Examiner further stated that “Bean discloses wherein a provider of remote web-site activity analysis (‘service provider’) generates JavaScript code that is distributed to each subscriber to the service and where the subscriber copies the code into each web-site page that is to be monitored (col. 2, lines 23-29).” Final Office Action, p. 11. Applicants assume the Examiner is referring to the description found at col. 1, lines 23-29 rather than col. 2, lines 23-29. As discussed above, this technique of generating code and copying it into web pages is well known in the art, but does not provide the unique advantages conferred by the claimed invention. Specifically, such conventional techniques fail to provide any method for tracking custom events, since the code is standard and not customized. As stated in Bean, the information passed by such code is predetermined. As such, there is no mention anywhere in Bean of any steps of modifying tracking code to track occurrence of a custom event, as claimed herein.

In response to Applicants’ earlier arguments that Bean fails to teach the tracking of custom events, the Examiner stated that “Bean discloses when a visitor to the subscriber’s web site loads one of the website pages into his or her computer, the JavaScript code collects information, including time of day, visitor domain, page visited, etc; and in addition Bean teaches wherein this arrangement for monitoring web

server activity is known in the art (col. 1, lines 23-54). Furthermore, Bean discloses wherein the data mining code embedded within the web page script operates to gather data about the visitor's computer. Lastly, Bean discloses another method for tracking visitors to a web site through the use of objects called cookies (col. 3, lines 7-35)." Final Office Action, p. 12.

With respect, Applicants respectfully point out that the Examiner has mischaracterized the cited reference. Bean teaches JavaScript code that collects information about the user and the site visit. Again, such techniques are well known in the art. However, what are missing from Bean are the specific steps (recited in the claims herein) for tracking custom events. Such steps include receiving user input for selecting customizable events, receiving user input for selecting custom attributes, modification of embedded tracking code (not merely data, but code) to track the specified custom events, receiving user input specifying configuration of a data collection server to receive custom events (including specified values for an event and a custom attribute), configuration of a data collection server to receive custom events, and generation of a configuration string for inclusion on a web page. Of all these steps that are explicitly recited in claim 1, not a single one is taught by Bean.

Bean describes the use of JavaScript code merely for collecting information and transmitting it to a server, as is well known. (Col. 1, lines 23-54). Bean's data mining for gathering data and storing client data in a cookie are also well known, and merely constitute accepted methods for collecting and storing information about

a web visitor. (Col. 3, lines 7-35). However, nowhere in these cited portions are any of the above-mentioned steps of claim 1 described or taught.

In response to Applicants' earlier arguments that Bean's description of changing name-value pairs is not equivalent to modification of embedded tracking code, the Examiner stated that "the fact that 'name-value pairs' can be changed and stored or embedded as pieces of information that make up a 'cookie' for the purpose of tracking a user's visit to the website, discloses the claimed limitation." Final Office Action, p. 12-13. Applicants respectfully point out that the Examiner is improperly conflating the notion of modifying data with modification of software code. Changing, storing, or embedding name-value pairs are merely processes of data storage and data update. Such processes cannot be said to be equivalent to, or to teach, modification of tracking code as recited herein. The fact that such name-value pairs contain data that may be used for tracking visits is entirely irrelevant; they are still data and not code. The claim, on the other hand, recites modifying embedded tracking code, and Bean's use of name-value pairs to store data simply does not teach such a step.

In addition, there is no hint or suggestion anywhere in Bean of any mechanism for receiving user input specifying configuration of a data collection server to receive a custom event, nor for generating a configuration string wherein the configuration string comprises encoded configuration information for at least one custom attribute, as claimed herein.

Therefore, Applicants respectfully submit that claim 1, as amended, contains several limitations that Bean fails to teach or suggest. Claim 1 is hereby submitted to be patentably distinct from the cited reference. Applicants respectfully request that the 102 rejection be withdrawn.

Claims 2-6 depend from claim 1 and incorporate the limitations discussed above. Therefore, for at least the reasons discussed above, claims 2-6 are submitted to be patentably distinct from the cited reference.

Claims 2-6 recite additional limitations not found in the cited reference. For example, claim 2 recites “modifying the embedded tracking code to associate at least one selected custom attribute with at least one selected custom event.” Nowhere in Bean is there any mention of modifying code in this manner, to associate a custom attribute with a selected custom event. The Examiner cited portions of Bean that describe changing of name-value pairs, storing IDs in cookies, and customization capability, none of which has any relation to associating a custom attribute with a custom event. The Examiner further cited col. 9, line 34 to col. 10, line 10 of Bean, which merely describe a method for conditionally including data mining code in data sent to a visitor computer; however, there is still no mention of modifying embedded tracking code in the manner recited in claim 2.

As another example, claim 4 recites assigning expiration data to at least one custom attribute, wherein the configuration string comprises a representation of the expiration data. Claim 5 recites assigning version data to at least one custom

attribute, wherein the configuration string comprises a representation of the version data. Nowhere in Bean is there any mention of a configuration string that comprises a representation of expiration or version data.

As another example, claim 6 recites inserting the configuration string in data collection code on the web page. Bean does not use a configuration string, and therefore fails to recite any method for inserting a configuration string in data collection code. The Examiner cited col. 3, lines 5-17 and col. 5, lines 6-13. However, col. 3, lines 5-17 merely describe general operation of data mining code embedded in a web page script. There is a reference to a “code string”. However, this code string represents data mined from the visitor computer, and is not a configuration string or any equivalent thereof. Col. 5, lines 6-13 merely describe storage of user preferences (customization information) by the well known technique of storing location or visitor information in a name-value pair of a cookie file. However, such customization information is used for changing the presentation of the web page to the user, and is not in any way equivalent to a configuration string for data collection. The present claims explicitly recite the use of the configuration string for inclusion on the web page and encoding configuration information for at least one custom attribute associated with at least one customizable event. This is entirely distinct from Bean’s storage of name-value pairs for customizing visual aspects of the web page to be displayed to the user.

Claims 17-20 recite one or more computer readable storage devices having computer readable code embodied thereon, and include limitations similar to those discussed above in connection with claim 1. Therefore, for at least the reasons discussed above, claims 17-20 are submitted to be patentably distinct from the cited reference.

New claims 21-22 depend from independent claims 1 and 17, respectively, and incorporate all of the limitations of the respective independent claims. Therefore, for at least the reasons discussed above, claims 21-22 are submitted to be patentably distinct from the cited reference.

Support for the claim amendments and new claims can be found in the originally filed specification at, for example, paragraphs [0022], [0023], [0052], [0066], [0068] to [0070], and Fig. 4. No new matter has been added.

On the basis of the above amendments, consideration of this application and the early allowance of all claims herein are requested.

Should the Examiner wish to discuss the above amendments and remarks, or if the Examiner believes that for any reason direct contact with Applicants' representative would help to advance the prosecution of this case to finality, the Examiner is invited to telephone the undersigned at the number given below.

Respectfully submitted,
Brett Error, et al.

Dated: July 30, 2009

By: / Amir H. Raubvogel/
Amir H. Raubvogel
Reg. No. 37,070
Raubvogel Law Office
820 Lakeview Way
Redwood City, CA 94062
Phone: (650) 209-4884
Fax: (650) 362-1800